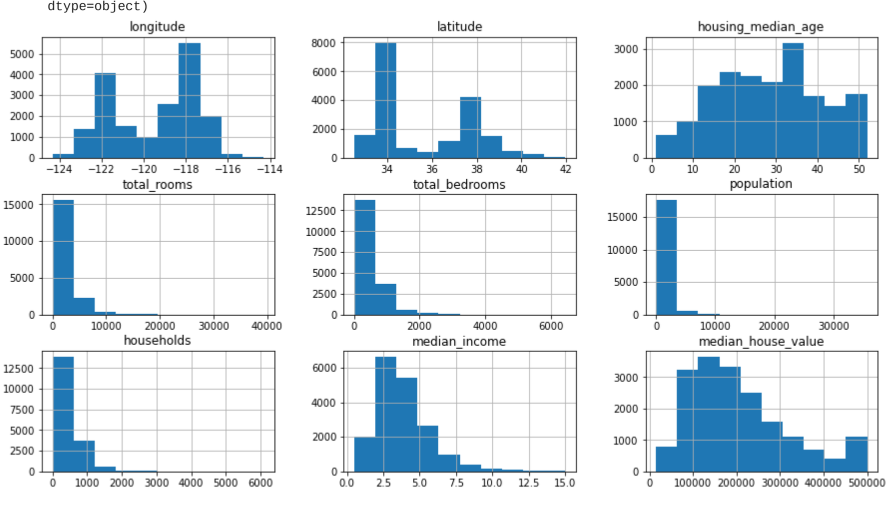
In [228…

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv("housing.csv")

print(data)

#Check the dataset
data.info()

#Removing the null values
data.dropna(inplace=True)

#Check the data without the null values
data.info()

from sklearn.model_selection import train_test_split
#Split the dataset in X & y
X = data.drop(['median_house_value'] , axis=1)

y = data['median_house_value']

print(X)
print(y)

X_train , X_test, y_train, y_test = train_test_split(X, y, test_size=0.1)
#Join two training dataset
train_data = X_train.join(y_train)

print(train_data)

train_data.hist(figsize=(15, 8))
```

```
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0        -122.23     37.88                41.0        880.0           129.0
1        -122.22     37.86                21.0       7099.0          1106.0
2        -122.24     37.85                52.0       1467.0           190.0
3        -122.25     37.85                52.0       1274.0           235.0
4        -122.25     37.85                52.0       1627.0           280.0
...          ...       ...                 ...          ...             ...
20635    -121.09     39.48                25.0       1665.0           374.0
20636    -121.21     39.49                18.0        697.0           150.0
20637    -121.22     39.43                17.0       2254.0           485.0
20638    -121.32     39.43                18.0       1860.0           409.0
20639    -121.24     39.37                16.0       2785.0           616.0

       population  households  median_income  median_house_value  \
0           322.0       126.0         8.3252            452600.0
1          2401.0      1138.0         8.3014            358500.0
2           496.0       177.0         7.2574            352100.0
3           558.0       219.0         5.6431            341300.0
4           565.0       259.0         3.8462            342200.0
...           ...         ...            ...                 ...
20635       845.0       330.0         1.5603             78100.0
20636       356.0       114.0         2.5568             77100.0
20637      1007.0       433.0         1.7000             92300.0
20638       741.0       349.0         1.8672             84700.0
20639      1387.0       530.0         2.3886             89400.0

      ocean_proximity
0            NEAR BAY
1            NEAR BAY
2            NEAR BAY
3            NEAR BAY
4            NEAR BAY
...               ...
20635          INLAND
20636          INLAND
20637          INLAND
20638          INLAND
20639          INLAND

[20640 rows x 10 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20433 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20433 non-null  float64
 1   latitude            20433 non-null  float64
 2   housing_median_age  20433 non-null  float64
 3   total_rooms         20433 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20433 non-null  float64
 6   households          20433 non-null  float64
 7   median_income       20433 non-null  float64
 8   median_house_value  20433 non-null  float64
 9   ocean_proximity     20433 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.7+ MB
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
0        -122.23     37.88                41.0        880.0           129.0
1        -122.22     37.86                21.0       7099.0          1106.0
2        -122.24     37.85                52.0       1467.0           190.0
3        -122.25     37.85                52.0       1274.0           235.0
4        -122.25     37.85                52.0       1627.0           280.0
...          ...       ...                 ...          ...             ...
20635    -121.09     39.48                25.0       1665.0           374.0
20636    -121.21     39.49                18.0        697.0           150.0
20637    -121.22     39.43                17.0       2254.0           485.0
20638    -121.32     39.43                18.0       1860.0           409.0
20639    -121.24     39.37                16.0       2785.0           616.0

       population  households  median_income ocean_proximity
0           322.0       126.0         8.3252        NEAR BAY
1          2401.0      1138.0         8.3014        NEAR BAY
2           496.0       177.0         7.2574        NEAR BAY
3           558.0       219.0         5.6431        NEAR BAY
4           565.0       259.0         3.8462        NEAR BAY
...           ...         ...            ...             ...
20635       845.0       330.0         1.5603          INLAND
20636       356.0       114.0         2.5568          INLAND
20637      1007.0       433.0         1.7000          INLAND
20638       741.0       349.0         1.8672          INLAND
20639      1387.0       530.0         2.3886          INLAND

[20433 rows x 9 columns]
0        452600.0
1        358500.0
2        352100.0
3        341300.0
4        342200.0
           ...
20635     78100.0
20636     77100.0
20637     92300.0
20638     84700.0
20639     89400.0
Name: median_house_value, Length: 20433, dtype: float64
       longitude  latitude  housing_median_age  total_rooms  total_bedrooms  \
9849     -121.89     36.60                40.0        626.0           164.0
19895    -119.21     36.30                23.0        951.0           235.0
15551    -117.09     33.11                32.0       1713.0           321.0
13444    -117.43     34.08                13.0       4563.0          1187.0
14559    -117.22     32.83                34.0       2936.0           597.0
...          ...       ...                 ...          ...             ...
2903     -119.02     35.39                52.0        191.0            52.0
7837     -118.16     33.91                 6.0       3445.0           847.0
898      -121.96     37.53                18.0       2375.0           652.0
4320     -118.35     34.09                47.0       1800.0           546.0
1013     -121.76     37.68                52.0       2157.0           418.0

       population  households  median_income ocean_proximity  \
9849        337.0       150.0         2.7917       <1H OCEAN
19895       806.0       222.0         1.7734          INLAND
15551       891.0       286.0         3.1429       <1H OCEAN
13444      2475.0      1019.0         2.1189          INLAND
14559      1512.0       571.0         3.7841      NEAR OCEAN
...           ...         ...            ...             ...
2903        106.0        49.0         2.0455          INLAND
7837       2467.0       712.0         3.1507       <1H OCEAN
898        1252.0       586.0         2.6198       <1H OCEAN
4320        921.0       478.0         2.8021       <1H OCEAN
1013        929.0       419.0         3.7301          INLAND

       median_house_value
9849             225000.0
19895             41400.0
15551            171600.0
13444            121700.0
14559            176900.0
...                   ...
2903              72500.0
7837             144000.0
898              235900.0
4320             280600.0
1013             204400.0

[18389 rows x 10 columns]
```

Out[228…

```
array([[<AxesSubplot:title={'center':'longitude'}>,
        <AxesSubplot:title={'center':'latitude'}>,
        <AxesSubplot:title={'center':'housing_median_age'}>],
       [<AxesSubplot:title={'center':'total_rooms'}>,
        <AxesSubplot:title={'center':'total_bedrooms'}>,
        <AxesSubplot:title={'center':'population'}>],
       [<AxesSubplot:title={'center':'households'}>,
        <AxesSubplot:title={'center':'median_income'}>,
        <AxesSubplot:title={'center':'median_house_value'}>]],
      dtype=object)
```



In [229…

```python
train_data.corr()
```

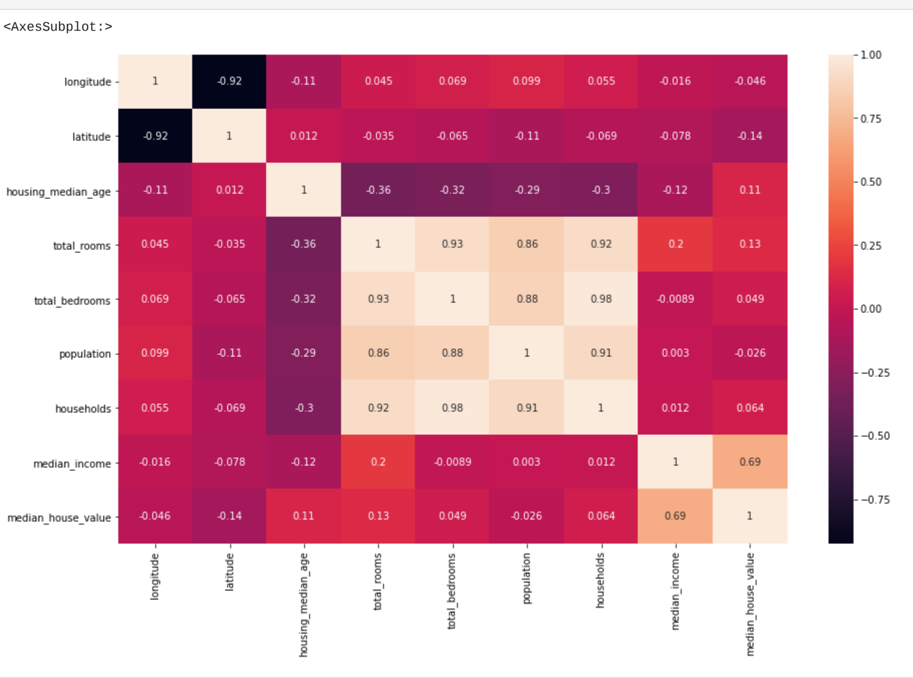
Out[229…

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
longitude	1.000000	-0.924659	-0.110364	0.044934	0.069081	0.098616	0.055457	-0.015690	-0.045837
latitude	-0.924659	1.000000	0.012232	-0.034507	-0.064896	-0.106267	-0.069368	-0.078281	-0.143646
housing_median_age	-0.110364	0.012232	1.000000	-0.358188	-0.319035	-0.292461	-0.301045	-0.117591	0.107674
total_rooms	0.044934	-0.034507	-0.358188	1.000000	0.931366	0.856322	0.919979	0.195721	0.131091
total_bedrooms	0.069081	-0.064896	-0.319035	0.931366	1.000000	0.877028	0.979519	-0.008943	0.048620
population	0.098616	-0.106267	-0.292461	0.856322	0.877028	1.000000	0.906876	0.003029	-0.026126
households	0.055457	-0.069368	-0.301045	0.919979	0.979519	0.906876	1.000000	0.011829	0.064105
median_income	-0.015690	-0.078281	-0.117591	0.195721	-0.008943	0.003029	0.011829	1.000000	0.688668
median_house_value	-0.045837	-0.143646	0.107674	0.131091	0.048620	-0.026126	0.064105	0.688668	1.000000

In [230…

```python
plt.figure(figsize=(15,9))
sns.heatmap(train_data.corr(),annot=True)
```

Out[230… <AxesSubplot:>



In [231…

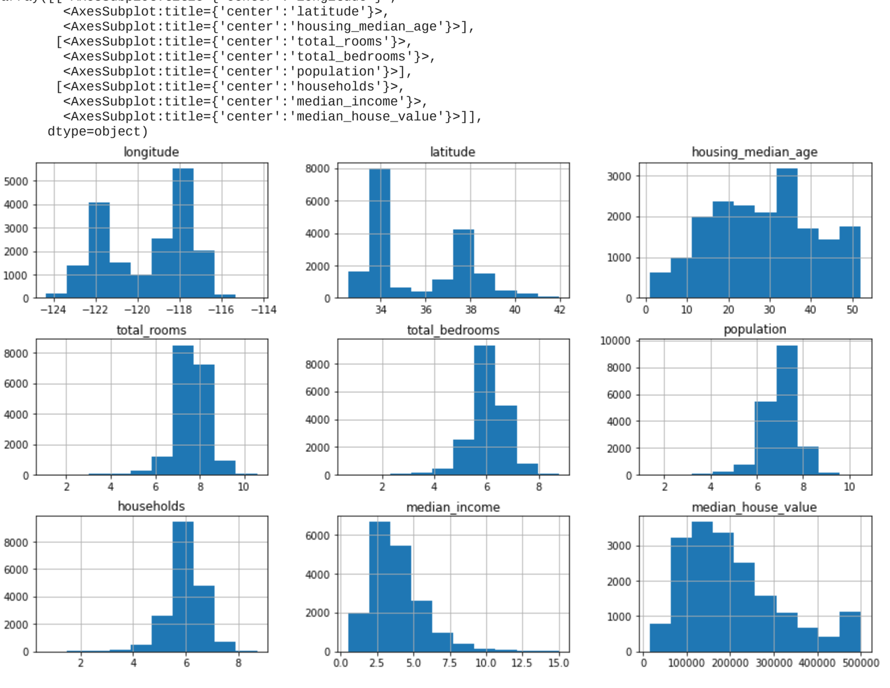
```python
train_data['total_rooms'] = np.log(train_data['total_rooms'] +1)
train_data['total_bedrooms'] = np.log(train_data['total_bedrooms'] +1)
train_data['population'] = np.log(train_data['population'] +1)
train_data['households'] = np.log(train_data['households'] +1)
```

In [232…

```python
train_data.hist(figsize=(15, 9))
```

Out[232…

```
array([[<AxesSubplot:title={'center':'longitude'}>,
        <AxesSubplot:title={'center':'latitude'}>,
        <AxesSubplot:title={'center':'housing_median_age'}>],
       [<AxesSubplot:title={'center':'total_rooms'}>,
        <AxesSubplot:title={'center':'total_bedrooms'}>,
        <AxesSubplot:title={'center':'population'}>],
       [<AxesSubplot:title={'center':'households'}>,
        <AxesSubplot:title={'center':'median_income'}>,
        <AxesSubplot:title={'center':'median_house_value'}>]],
      dtype=object)
```



In [233…

```python
train_data.ocean_proximity.value_counts()
```

Out[233…

```
<1H OCEAN     8124
INLAND        5836
NEAR OCEAN    2385
NEAR BAY      2040
ISLAND           4
Name: ocean_proximity, dtype: int64
```

In [234…

```python
train_data = train_data.join(pd.get_dummies(train_data.ocean_proximity)).drop(['ocean_proximity'], axis=1)
```

In [235…

```python
train_data
```

Out[235…

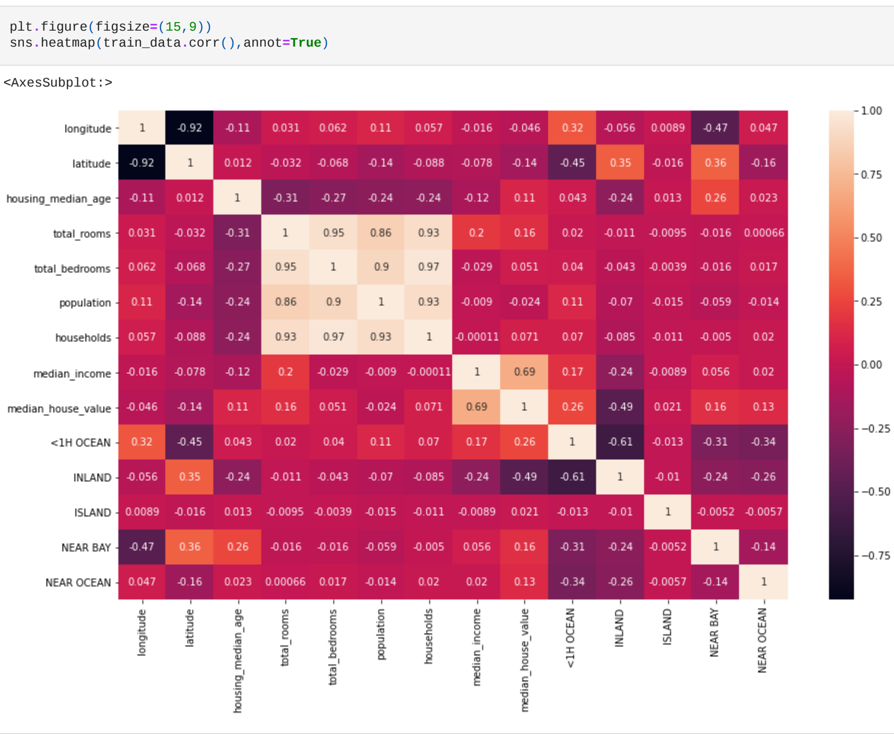
	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	<1H OCEAN	INLAND	ISLAND	NEAR BAY	NEAR OCEAN
9849	-121.89	36.60	40.0	6.440947	5.105945	5.823046	5.017280	2.7917	225000.0	1	0	0	0	0
19895	-119.21	36.30	23.0	6.858565	5.463832	6.693324	5.407172	1.7734	41400.0	0	1	0	0	0
15551	-117.09	33.11	32.0	7.446585	5.774552	6.793466	5.659482	3.1429	171600.0	1	0	0	0	0
13444	-117.43	34.08	13.0	8.425955	7.080026	7.814400	6.927558	2.1189	121700.0	0	1	0	0	0
14559	-117.22	32.83	34.0	7.985144	6.393591	7.321850	6.349139	3.7841	176900.0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2903	-119.02	35.39	52.0	5.257495	3.970292	4.672829	3.912023	2.0455	72500.0	0	1	0	0	0
7837	-118.16	33.91	6.0	8.144969	6.742881	7.811163	6.569481	3.1507	144000.0	1	0	0	0	0
898	-121.96	37.53	18.0	7.773174	6.481577	7.133296	6.375025	2.6198	235900.0	1	0	0	0	0
4320	-118.35	34.09	47.0	7.496097	6.304449	6.826545	6.171701	2.8021	280600.0	1	0	0	0	0
1013	-121.76	37.68	52.0	7.676937	6.037871	6.835185	6.040255	3.7301	204400.0	0	1	0	0	0

18389 rows × 14 columns

In [236…

```python
plt.figure(figsize=(15,9))
sns.heatmap(train_data.corr(),annot=True)
```

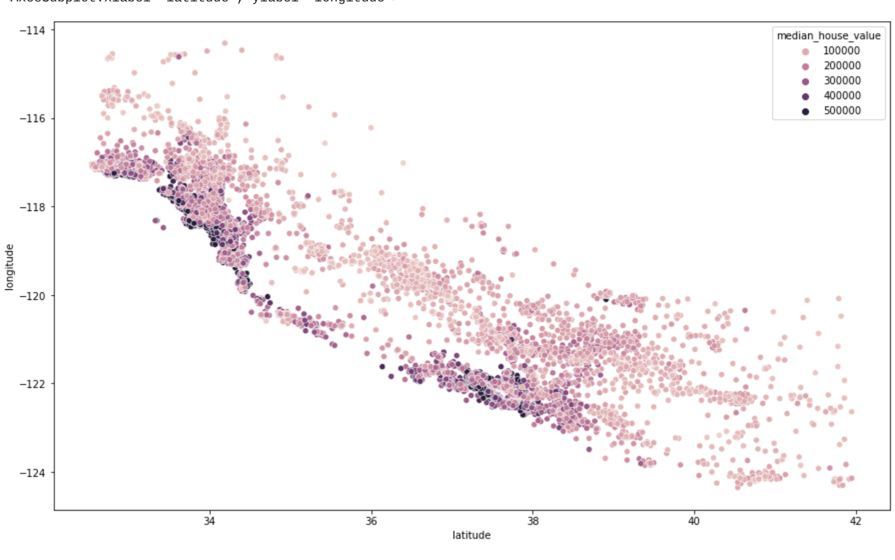
Out[236… <AxesSubplot:>



In [237…

```python
plt.figure(figsize=(15,9))
sns.scatterplot(x="latitude", y= "longitude", data=train_data, hue="median_house_value")
```

Out[237… <AxesSubplot:xlabel='latitude', ylabel='longitude'>



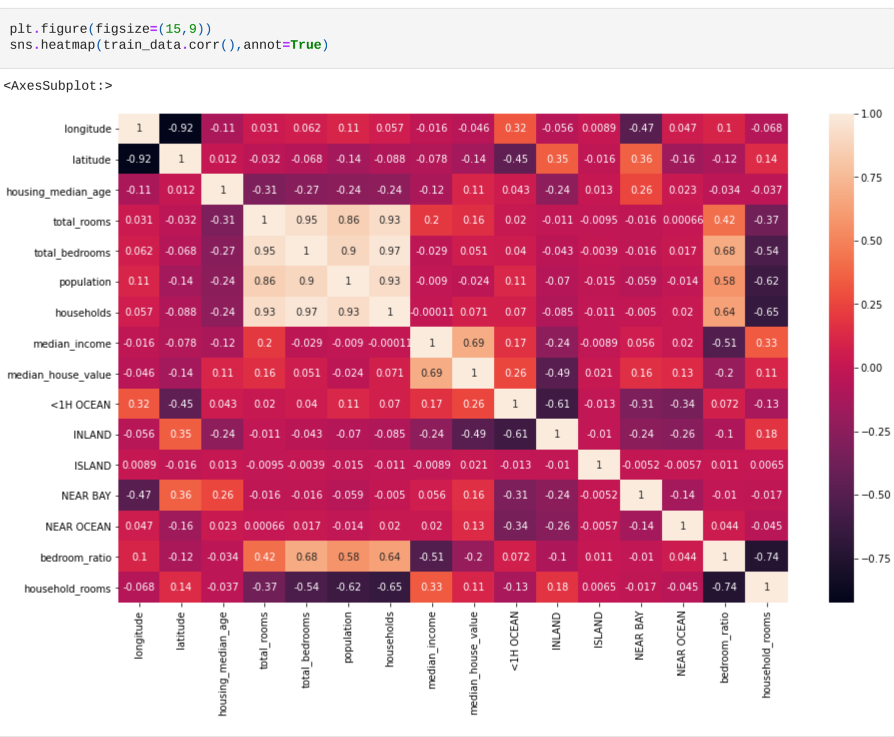
In [238…

```python
train_data['bedroom_ratio'] = train_data['total_bedrooms'] / train_data['total_rooms']
train_data['household_rooms'] = train_data['total_rooms'] / train_data['households']
```

In [239…

```python
plt.figure(figsize=(15,9))
sns.heatmap(train_data.corr(),annot=True)
```

Out[239… <AxesSubplot:>



In [240…

```python
from sklearn.linear_model import LinearRegression

X_train, y_train = train_data.drop(['median_house_value'], axis =1), train_data['median_house_value']

reg = LinearRegression()

reg.fit(X_train, y_train)
```

Out[240… LinearRegression()

In [243…

```python
test_data = X_test.join(y_test)

test_data['total_rooms'] = np.log(test_data['total_rooms'] +1)
test_data['total_bedrooms'] = np.log(test_data['total_bedrooms'] +1)
test_data['population'] = np.log(test_data['population'] +1)
test_data['households'] = np.log(test_data['households'] +1)

test_data = test_data.join(pd.get_dummies(test_data.ocean_proximity)).drop(['ocean_proximity'], axis=1)

test_data['bedroom_ratio'] = test_data['total_bedrooms'] / test_data['total_rooms']
test_data['household_rooms'] = test_data['total_rooms'] / test_data['households']


X_test, y_test = test_data.drop(['median_house_value'], axis =1), test_data['median_house_value']
```

In [244…

```python
test_data
```

Out[244…

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	<1H OCEAN	INLAND	ISLAND	NEAR BAY	NEAR OCEAN	bedroom_ratio	household_rooms
17630	-121.91	37.25	31.0	7.573017	5.840642	6.883463	5.814131	4.9205	240500.0	1	0	0	0	0	0.771244	1.302519
8877	-118.49	34.04	48.0	7.775696	5.846439	6.756932	5.726848	8.0257	500001.0	1	0	0	0	0	0.751886	1.357762
17575	-121.93	37.32	50.0	7.035269	5.375278	6.216606	5.337538	4.2614	211300.0	1	0	0	0	0	0.764047	1.318074
11147	-117.98	33.84	35.0	6.892642	5.192957	6.495266	5.298317	5.0747	189600.0	1	0	0	0	0	0.753406	1.300911
19235	-122.64	38.48	19.0	8.084871	6.109248	7.069023	6.120297	5.8369	255700.0	1	0	0	0	0	0.755639	1.320993
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1093	-121.89	39.71	26.0	7.916443	6.113682	7.104965	6.082219	3.7007	139200.0	0	1	0	0	0	0.772276	1.301572
16871	-122.41	37.60	31.0	8.395026	6.727432	7.557995	6.706862	4.1364	412000.0	0	0	0	0	1	0.801359	1.251707
14042	-117.14	32.76	28.0	8.014997	6.629363	7.192182	6.545350	2.6940	164100.0	0	0	0	0	1	0.827120	1.224533
9952	-122.32	38.35	20.0	8.159089	6.309918	7.422971	6.295266	5.5718	185200.0	0	1	0	0	0	0.773361	1.296067
5677	-118.31	33.67	42.0	7.168580	5.509388	6.416732	5.493061	5.3074	401900.0	0	0	0	0	1	0.768547	1.305025

2044 rows × 16 columns

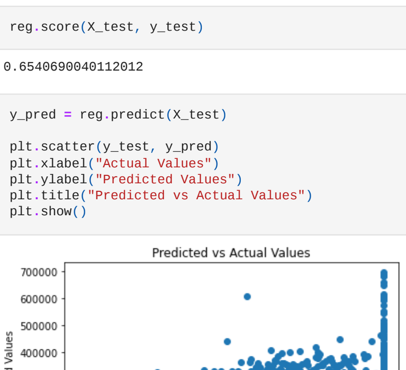
In [245…

```python
reg.score(X_test, y_test)
```

Out[245… 0.6540690040112012

In [262…

```python
y_pred = reg.predict(X_test)

plt.scatter(y_test, y_pred)
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("Predicted vs Actual Values")
plt.show()
```



In [254…

```python
from sklearn.ensemble import RandomForestRegressor

forest = RandomForestRegressor()

forest.fit(X_train, y_train)
```

Out[254… RandomForestRegressor()

In [257…

```python
RandomForest_prediction = forest.score(X_test, y_test)
```

In [258…

```python
RandomForest_prediction
```

Out[258… 0.9601078455493278

In [261…

```python
y_pred = forest.predict(X_test)

plt.scatter(y_test, y_pred)
plt.xlabel("Actual Values")
plt.ylabel("Predicted Values")
plt.title("Predicted vs Actual Values")
plt.show()
```